

**BARCOS EN CONTIENDA**

PROGRAMACIÓN ORIENTADA A OBJETOS

GRUPO 1 – EQUIPO 7

**ESTUDIANTES:**

Miguel Fernando Olave Riascos (*molave@unal.edu.co),* Camilo José Funez García *(cjfunezg@unal.edu.co),* Juan Diego Ortiz Tabares *(juortizt@unal.edu.co)*

**PROFESOR:**

Jaime Alberto Guzmán Luna

**UNIVERSIDAD NACIONAL DE COLOMBIA**

2021-II

**TABLA DE CONTENIDO**

1. Descripción general de la solución
2. Descripción del diseño estático del sistema en la especificación UML

2.1. Diagrama de Clases

1. Descripción de la implementación de características de programación orientada a objetos en el proyecto.
2. Descripción de cada una de las 5 funcionalidades implementadas.
3. Manual de usuario.

**1. Descripción general de la solución:**

El propósito de la realización de este proyecto es el de crear un videojuego adentrándose en la vida de un capitán de la marina remontada en el siglo XIX, aquí ocurrirán eventos que simulen enfrentamientos entre diferentes tipos de barcos, cada uno con sus propios atributos, haciendo que cada barco en particular llegue a tener características diferentes al resto de los demás barcos. De igual manera, se crearán métodos para generar interacciones entre estos barcos, e incluso interacciones entre los barcos y elementos neutros, como Islas, Puertos, y Puntos de Venta. Por último, también juegan un papel muy importante las características (atributos) de dichos barcos y demás clases generando buffs y nerfs aleatoriamente para dichas clases.

**2. Descripción del diseño estático del sistema en la especificación UML:**

**2.1. Diagrama de Clases:**

Diagrama, Esquemático

Descripción generada automáticamente

Se realizaron 6 clases diferentes, una para cada aspecto principal del desarrollo del videojuego. Tales son **Barco**, **Capitanes**, **Tripulación**, **Lugar**, **Botín** y **Objeto**.

De la clase Capitanes se heredan 4 subclases, las cuales especifican el tipo de Capitán que se puede presentar en el videojuego, los cuales son **CapitanPirata**, **CapitanFantasma**, **CapitanCivil** y **CapitanNaval**; de igual forma, de la clase Lugar se hereda **Isla** y **Puerto.**

Los objetos principales se crearán a partir de las clases de **Barco**, **Capitanes**, **Lugar**, **Tripulacion** y **Botin.**

1. **Descripción de la implementación de características de programación orientada a objetos en el proyecto:**

Se implementaron múltiples elementos de la programación orientada a objetos en el desarrollo del proyecto. En primera instancia, tenemos la clase **Barco**, que es de las clases principales que se realizaron para la ejecución de la idea. Esta tiene múltiples atributos y métodos, que de igual forma se pueden observar en el diagrama de clases. Los métodos más importantes en esta clase son los que sirven como nexo de la ejecución del videojuego, y se tratan de los que son de apoyo para la interacción entre los barcos que pueden existir en el videojuego.

La creación de muchos de estos barcos, que son con los que el usuario va a interactuar una vez esté el videojuego en ejecución, tomaron valores aleatorios para la creación de ciertos atributos, como, por ejemplo, la **elocuencia** de los demás capitanes, la puntuación de **ataque** de los barcos que existen, o la cantidad de tripulantes que hay en cierta facción del videojuego.

Se tiene, en la clase de **Botin**, existe una sobrecarga de constructores:

Texto

Descripción generada automáticamente

Esto resulta útil para adaptarse a las necesidades de la ejecución del código, Ya que por un lado, se puede añadir al **Botin** una lista completa de objetos, o solamente un objeto. También, sirve para indicar la cantidad existente de oro en el **Botin**, de tal forma que dependiendo del **Botín** resultante al hacer un ataque a un **Barco**, este pueda ser añadido a la lista de **Botin** del **Barco** del usuario.

También, hay un aspecto muy importante, y es la existencia de la **Herencia de Clases**, específicamente, de la clase padre **Capitanes**. De esta, se heredan las clases **CapitanCivil**, **CapitanFantasma**, **CapitanNaval** y **CapitanPirata**. Lo particular de estas clases hijas es que sirven para diferenciar la facción de los tipos de capitanes, y por ende, de los propios barcos. Esto es importante ya que de acuerdo a ciertas facciones, los barcos podrían tener más o menos defensa, cantidad de tripulantes, o capitanes con mejor liderazgo.

Texto

Descripción generada automáticamente

En algunas clases se pueden evidenciar atributos de tipo **private**, ya que, en sí, son exclusivos de esa propia clase en la que se está trabajando, por ejemplo, el **Lugar** en el que está asentada una tripulación, sus puntuaciones de **Ataque** y **Defensa**.

Texto

Descripción generada automáticamente

Además, se implementa la creación de clases y métodos abstractos, específicamente en la clase **Capitanes**, que sirve para unificar el resto de las subclases de esta misma. El método abstracto que se crea en esta clase es el de **yells()**, el cual sirve para identificar el tipo de capitán al cual se está haciendo referencia dentro de la ejecución del videojuego.

Texto

Descripción generada automáticamente

Incluso se implementa la **enumeración**, en la clase **Faccion**, lo cual llega a ser fundamental para poder diferenciar entre facciones de barcos, y como fue dicho anteriormente, poder realizar diferentes interacciones entre estos barcos y tener diferentes atributos o características.

Texto

Descripción generada automáticamente

Elementos como **this** se pueden evidenciar, por ejemplo, en la clase **Puerto**, donde en varios métodos se aplica esta implementación para determinar explícitamente a qué se está haciendo referencia.

Texto

Descripción generada automáticamente

1. **Descripción de cada una de las 5 funcionalidades implementadas:**

* **Compra de mercancía:** El usuario podrá “comprar” la mercancía que pueda encontrar en diferentes lugares, por ejemplo, en los puertos. Obtener baratijas u objetos que le resultarán útiles a la hora de realizar más expediciones por el mundo.
* **Personalización de barcos:** El usuario podrá personalizar varios aspectos de su barco, como, por ejemplo, el nombre de este, su capitán, contratar/comprar tripulantes para tener mayor ventaja a la hora de realizar asaltos e incursiones, e incluso reparar su propio barco, para restaurar sus valores de defensa y ataque.
* **Asalta otros barcos:** El usuario podrá realizar asaltos a otros barcos, esto con el fin de apoderarse de su botín y aumentar la riqueza de su barco, también, para conseguir objetos raros y útiles para después preparar asaltos a islas misteriosas en las cuales le podrá esperar un botín muy valioso.
* **Ataca a otros barcos:** El usuario podrá realizar ataques a otros barcos, esto con el fin de que, una vez debilitado, apoderarse de su botín y aumentar la riqueza de su barco; también, para conseguir objetos raros y útiles para después preparar asaltos a islas misteriosas en las cuales le podrá esperar un botín muy valioso.
* **Asaltos a islas:** El usuario podrá preparar y realizar asaltos a otras islas, las cuales podrán ser habitadas por otras tripulaciones para proteger sus riquezas, o incluso encontrar mapas del tesoro y llaves para abrir cofres en otras islas, los cuales tendrán recompensas muy valiosas.

A continuación, podrán observarse ejemplos de la ejecución del videojuego, en el cual se evidenciarán los diferentes caminos a escoger por parte del usuario.

En primer lugar, el usuario puede personalizar el nombre de su barco y su capitán con total libertad.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Después, se le presentarán diferentes caminos a escoger, los cuales servirán para adentrarse en un combate naval, o inspeccionar las características de su barco.

Texto

Descripción generada automáticamente

Si el usuario escoge la opción de ir a un puerto (1), podrá realizar mejoras a su barco, o comprar mercancía para ingresarlas en su barco.

Texto

Descripción generada automáticamente

(**Aclaración**: La compra de un salvoconducto solo aplica para barcos piratas, el cual sirve para disminuir el nivel de búsqueda que tenga el barco en ese momento).

Si se escoge la opción de buscar un tipo específico de barco, ya sea civil (1), naval (2) o fantasma (3), el usuario podrá interactuar de manera hostil contra este, inspeccionarlo, o huir del combate después de recibir un saludo del otro barco y de una descripción de su facción.

Texto

Descripción generada automáticamente

Si el usuario decide asaltar una isla, comenzará un combate contra esta (el cual se dará con valores aleatorios, aunque tendrá más probabilidad de éxito si el barco del usuario tiene mejoras ya implementadas).

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Finalmente, el usuario puede inspeccionar las características de su barco para poder tomar futuras decisiones las cuales le den una ventaja respecto a los demás barcos o islas que pueda llegar a encontrarse.

Texto

Descripción generada automáticamente

1. **Manual de usuario:**

Para el correcto funcionamiento del videojuego, no se requiere de ningún tipo de dato o información externo, por lo cual, esto no aplica para el proyecto.